

Available online at www.sciencedirect.com**SciVerse ScienceDirect**

Procedia Computer Science 18 (2013) 1545 – 1554

Procedia
Computer Science

2013 International Conference on Computational Science

Hypergraph Grammars in hp-adaptive Finite Element Method

Grażyna Ślusarczyk^{a,*}, Anna Paszyńska^a^aJagiellonian University, Reymonta 4, 30-059 Kraków, Poland

Abstract

The paper presents the hypergraph grammar for modelling the hp-adaptive finite element method algorithm with rectangular elements. The finite element mesh is represented by a hypergraph. All mesh transformations are modelled by means of hypergraph grammar rules. These rules allow to generate the initial mesh, to assign values of polynomial order to the element nodes, to generate the matrix for each element, to solve the problem and to perform the hp-adaptation.

© 2013 The Authors. Published by Elsevier B.V. Open access under [CC BY-NC-ND license](#).

Selection and peer review under responsibility of the organizers of the 2013 International Conference on Computational Science

Keywords: finite element method; hypergraphs; hypergraph grammars.

1. Introduction

The paper presents an attempt to model the hp-adaptive Finite Element Method (hp-FEM) by means of a hypergraph grammar. The hp-FEM [1-3] consists in finding the solution of an engineering problem by constructing the sequence of approximation spaces, which approximate the solution in a more and more accurate way. The finite element mesh consists of finite elements and the shape functions corresponding to the finite element nodes, edges and interiors. The shape functions are glued together into global basis functions. In order to increase the accuracy of the solution, selected finite elements can be broken into smaller elements or the polynomial order of the approximation can be increased. The first attempt to model FEM by means of graph grammars has been presented in 1996 in [4]. In [10] the simple rewriting system based on topological chain rewriting was used to model mesh refinements. Both approaches model only uniform refinements. In order to model non-uniform refinements the context graph grammar should be used. In papers [7-9] the composite graph grammar has been used to model two and three dimensional hp-FEM. This paper presents the hypergraph grammar for two dimensional hp-FEM. The proposed approach allows to model mesh transformations: generation of the initial mesh as well as non-uniform refinements, in a more intuitive way than

* Corresponding author. Tel.: +48 -12- 663-56-98

E-mail address: gslusarc@uj.edu.pl.

the previous approaches. Using the hypergraph grammar instead of the composition graph grammar to model mesh transformations can decrease the computational complexity of performed operations, because the number of edges and nodes in the hypergraph representing a computational mesh is much smaller than the number of edges and nodes in the corresponding composite graph. In this paper a hypergraph representation of the finite element mesh is proposed. The topological structure of each mesh element is represented here using four hypergraph nodes corresponding to its vertices and five hyperedges corresponding to its edges and interior, while in the composition graph representation of a mesh element [7-9] 18 graph nodes with 60 node bonds are needed.

2. Hypergraph representation of the computational mesh

In [7-9] computational meshes have been represented by composition graphs. In this approach, to represent vertices, edges and interior of a mesh element, and relations between them many graph nodes with bonds are needed. The example of the structure of a finite element together with the corresponding composition graph is shown in Fig. 1 (node bonds are denoted as small circles inside labelled nodes). This representation is complex as nodes with the same labels should have the fixed number of bonds.

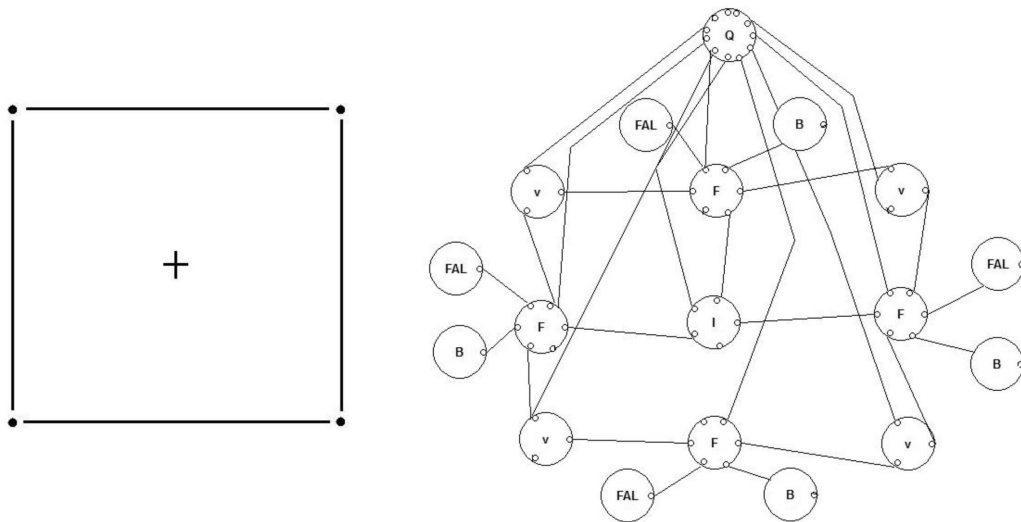


Fig. 1. A hypergraph representing an initial mesh

In this paper we propose to represent meshes in the form of hypergraphs [5-6]. This approach seems to be more intuitive as relations between mesh edges, interiors and vertices are represented by assigning nodes corresponding to vertices to hyperedges corresponding to edges and interiors.

Each hypergraph is composed of a set of nodes and a set of hyperedges with sequences of source and target nodes assigned to them. The nodes and hyperedges are labelled with the use of a fixed alphabet. When the undirected hypergraph is considered, the source nodes of each hyperedge are the same as its target nodes. To nodes and hyperedges the sets of attributes, which represent the properties of the corresponding mesh elements, are assigned. Additionally for each hypergraph a sequence of its external nodes is specified.

Let C be a fixed alphabet of labels for nodes and hyperedges. Let A be a set of hypergraph attributes.

Definition 1. An undirected attributed labelled hypergraph over C and A is a system $G = (V, E, t, l, at)$, where:

1. V is a finite set of nodes,
2. E is a finite set of hyperedges,
3. $t: E \rightarrow V^*$ is a mapping assigning sequences of target nodes to hyperedges of E ,
4. $l: V \cup E \rightarrow C$ is a node and hyperedge labelling function,
5. $at: V \cup E \rightarrow 2^A$ is a node and hyperedge attributing function.

Hypergraphs are created from simpler hypergraphs by replacing their hyperedges by new hypergraphs. This operation is possible if for each new hypergraph a sequence of its external nodes is specified. These nodes correspond to target nodes of a replaced hyperedge.

Definition 2. A hypergraph of type k is a system $H = (G, ext)$, where:

1. $G = (V, E, t, l, at)$ is a hypergraph over C and A ,
2. ext is a sequence of specified nodes of V , called external nodes, with $|ext| = k$.

In a hypergraph representation of a mesh, hypergraph nodes represent mesh vertices and are labeled by v , while hyperedges connecting these nodes represent interiors, edges and boundary edges of finite elements and are labelled by I , F and B , respectively.

An example of a hypergraph representing an initial mesh, which consists of one rectangular element, is shown in Fig. 2. It is composed of four nodes representing element vertices, and five hyperedges. The hyperedge with four target nodes and labelled I represents the mesh interior, while four hyperedges with two target nodes each and labelled B represent the boundary edges. Each node with label v and each hyperedge with label F or B has attribute p , which denotes the polynomial order of approximation. A hyperedge with label I has attributes p_h and p_v which denote polynomial order of approximation in the horizontal and vertical direction, respectively. Additionally, the hyperedge with label I has attributes m , s and err which denote the matrix, the solution and the vector of error for the corresponding element. The attribute hp , which is also assigned to the hyperedge labelled I denotes the kind of hp-adaptation and can be equal to one of the values from the set $\{0, 1, \dots, 24\}$. The attribute value $hp=0$ means no adaptation. The meaning of other values of attribute hp is presented in Fig. 3. The kinds of adaptation are numbered from 1 to 24, row by row.

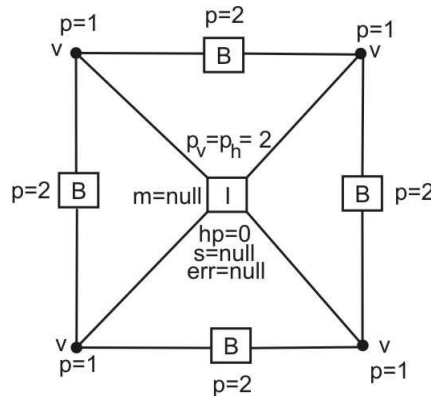


Fig. 2. A hypergraph representing an initial mesh

3. Hypergraph grammars for hp-adaptive finite element method

Hypergraphs are derived from other hypergraphs by replacing their subhypergraphs by new hypergraphs. The described rewriting method is a generalization of a hyperedge replacement mechanism presented in [5].

Hypergraphs are generated by applying productions, which specify the way of subhypergraph replacement.

Definition 3. A hypergraph production is a pair $p = (L, R)$, where both L and R are hypergraphs of the type k .

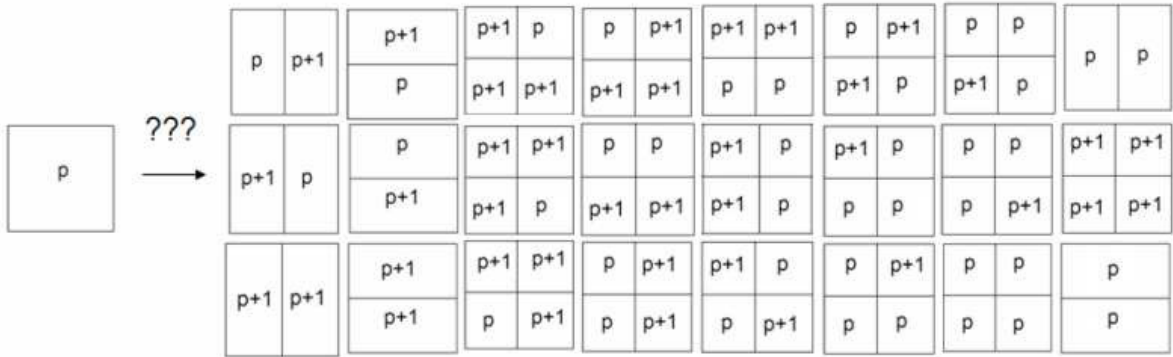


Fig. 3. Possible kinds of hp-adaptation

A production p can be applied to a hypergraph H if H contains a subhypergraph isomorphic with L .

Definition 4. Let $G_1 = (V_1, E_1, t_1, l_1, at_1)$ and $G_2 = (V_2, E_2, t_2, l_2, at_2)$ be two hypergraphs. G_1 is a subhypergraph of G_2 if:

1. $V_1 \subseteq V_2, E_1 \subseteq E_2$,
2. $\forall e \in E_1 t_1(e) = t_2(e)$,
3. $\forall e \in E_1 l_1(e) = l_2(e), \forall v \in V_1 l_1(v) = l_2(v)$,
4. $\forall e \in E_1 at_1(e) = at_2(e), \forall v \in V_1 at_1(v) = at_2(v)$.

The application of a production $p = (L, R)$ to a hypergraph H consist of replacing a subhypergraph of H isomorphic with L by a hypergraph R and replacing nodes of the removed subhypergraph isomorphic with external nodes of L by the corresponding external nodes of R .

The production which enables mesh generation is presented in Fig. 4a. Its left-hand side (hypergraph L) is composed of one hyperedge labelled B and two nodes being external ones. The hypergraph of the right-hand side (R) is composed of five hyperedges representing the interior (label I), boundary edges (labels B) and vertical edge (label FI), and four nodes representing mesh vertices. Nodes with numbers are external ones. Thus L and R are hypergraphs of type 2. The application of this production to the hyperedges labelled B of the initial hypergraph (Fig. 2) can result in the development of the mesh to the right or to the left. The mesh development to the top and to the bottom is realized by the analogical production, where the hyperedge labelled FI is replaced by the hyperedge with label $F2$ representing the horizontal edge. The result of one application of the production to the right B hyperedge of the initial hypergraph is shown in Fig. 4b.

Definition 5. Let P be a fixed set of hypergraph productions. Let H and H' be two hypergraphs. H' is directly derived from H ($H \Rightarrow H'$) if there exists $p = (L, R) \in P$ such that:

1. h is a subhypergraph of H isomorphic with L ,
2. Let ext_h be a sequence of nodes of h composed of nodes isomorphic with nodes of the sequence ext_L .

The replacement of $h = (V_h, E_h, t_h, l_h, at_h)$ in $H = (V_H, E_H, t_H, l_H, at_H)$ by $R = (V_R, E_R, t_R, l_R, at_R)$ yields the hypergraph $G = (V_G, E_G, t_G, l_G, at_G)$, where:

- $V_G = V_H - V_h \cup V_R$,
- $E_G = E_H - E_h \cup E_R$,
- $\forall e \in E_R t_G(e) = t_R(e)$,

- $\forall e \in E_H - E_h$ with $t_H(e) = t_1, \dots, t_n$, $t_G(e) = t_1', \dots, t_n'$, where each $t_i' = t_i$ if t_i does not belong to the sequence ext_h or $t_i' = v_j$ (v_j is a j -th element of the sequence ext_R) if t_i is an j -th element of the sequence ext_h ,
 - $\forall e \in E_H - E_h$ $l_G(e) = l_H(e)$, $at_G(e) = at_H(e)$, $\forall e \in E_R$ $l_G(e) = l_R(e)$, $at_G(e) = at_R(e)$,
 - $\forall v \in V_H - V_h$ $l_G(v) = l_H(v)$, $at_G(v) = at_H(v)$ $\forall v \in V_R$ $l_G(v) = l_R(v)$, $at_G(v) = at_R(v)$.
3. H' is isomorphic with the result of replacing h in H by R .

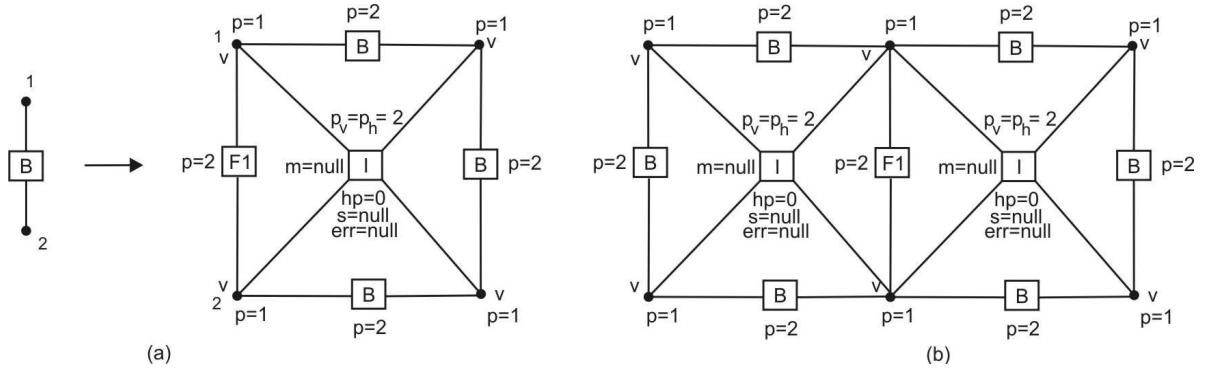


Fig. 4. (a) A hypergraph production; (b) a hypergraph representing a mesh composed of two elements

Let $AT = V \cup E$ be a set of nodes and hyperedges of \mathcal{H} , where \mathcal{H} denotes a family of hypergraphs over C and A .

Definition 6. A hypergraph grammar is a system $\mathcal{G} = (V, E, P, X)$, where:

1. V is a finite set of labelled nodes,
2. E is a finite set of labelled hyperedges,
3. P is a finite set of hypergraph productions of the form $p = (L, R)$, where L and R are hypergraphs of the same type composed of nodes of V and hyperedges of E ,
4. X is an initial hypergraph called axiom of \mathcal{G} .

The hypergraph grammar used to represent hp-adaptive finite element method is composed of four groups of productions:

1. productions describing the generation of the initial mesh (the production from Fig.4a)
2. productions for calculating the matrix, the solution and error for each finite element, as well as maximum relative error
3. productions for virtual hp-adaptation
4. productions performing hp-adaptation

The graph grammar for the hp-FEM performs all possible productions from the first and second group. If the maximum relative error of the solution is greater than the required accuracy, then all possible productions from the third and fourth group have to be applied, following by productions from the second group. Productions of each group are described beneath.

After generating the initial mesh (performing the production from Fig. 4a several times), the matrix for each element has to be calculated. Fig. 5 presents a production which to each element interior assigns the external calculated matrix.

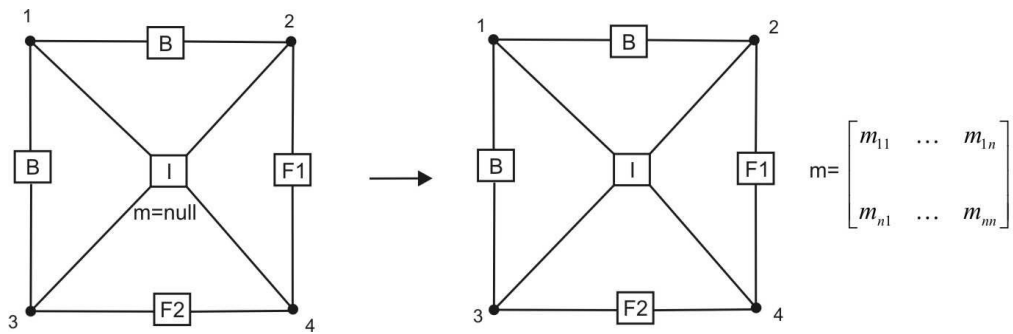


Fig. 5. A hypergraph production for calculating the matrix

The next step is calculating the solution and error for each element (see Fig. 6).

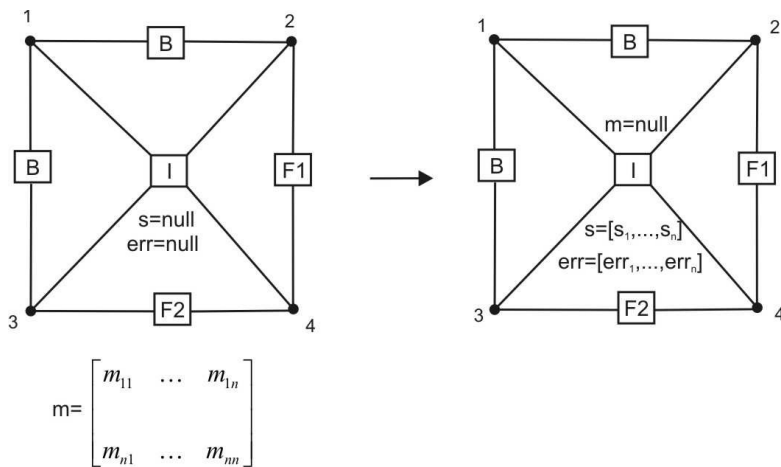


Fig. 6. A hypergraph production for calculating solution and error vector on the base of the matrix

After calculating the matrix, the solution vector and error vector for each element, the productions performing virtual hp-adaptation have to be applied. Fig. 7 presents the virtual hp-adaptation for the interior of the element. The decision about the kind of the adaptation is based on the error vector of the element and is denoted by $hp = f(err)$.

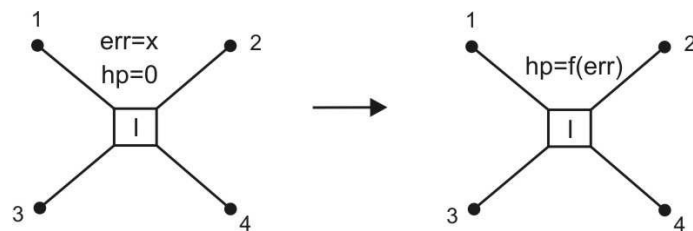


Fig. 7. A hypergraph production for virtual hp-adaptation

In order to follow the 1-irregularity rule (a finite element can be broken only once without breaking the adjacent large elements), after performing virtual hp-adaptation for each element, several additional refinements on large adjacent elements may be required. One of the productions propagating the adaptation is presented in Fig. 8. Similar production is defined for the hyperedge with label $F2$ instead of $F1$. The function $f(x,y)$ decides about the kind of the adaptation in the following way. If the value for hp attribute for bigger element equals 0 ($y=0$ - no adaptation) then the new value for hp attribute for the bigger element is the same as for the smaller element ($f(x,y)=x$). In other case $f(x,y)$ works like logical OR, for example if x denotes h-adaptation in one direction and y in other direction, then $f(x,y)$ denotes h-adaptation in both directions.

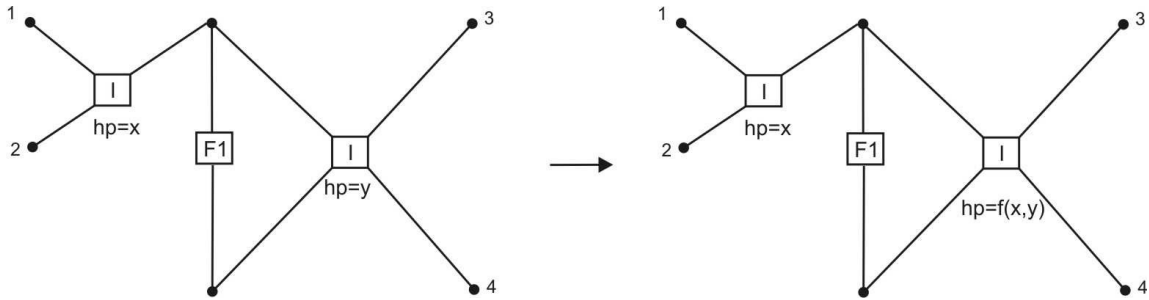


Fig. 8. A hypergraph production for propagating the adaptation

After performing productions for virtual hp-adaptation, the group of productions which performs hp-adaptation has to be used.

The hp-adaptation is modeled by:

- breaking interiors of elements and assigning values for attribute p for newly created interiors according to the value of attribute hp of the parental interior,
- breaking the edges,
- performing the minimum rule in order to calculate the polynomial order of approximation for edges (the value of attribute p).

Fig. 9 presents two productions for performing hp-adaptations, where hp equals to 12 and 1, respectively.

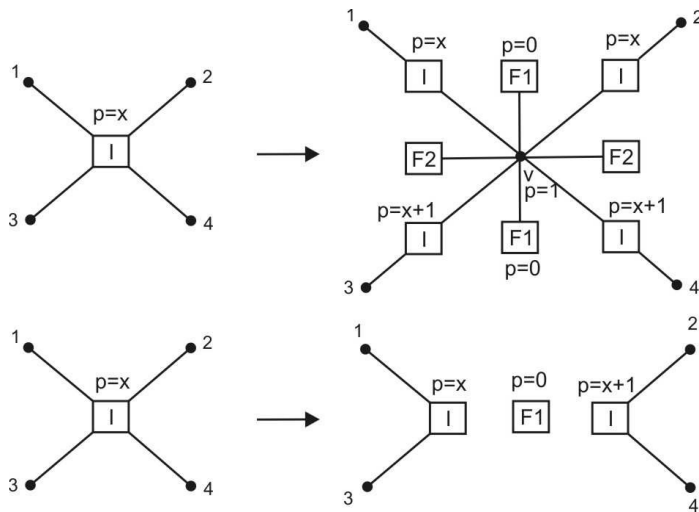


Fig. 9. Hypergraph productions for hp-adaptation

After breaking interiors of elements, all edges adjacent to two broken elements (Fig. 11) and boundary edges adjacent to one broken element (Fig. 10) have to be broken.

The last step of the hp-adaptation process is performing the minimum rule in order to calculate the polynomial order of approximation for edges (the value of attribute p). Productions which assign values for attribute p for edges are presented in Fig. 12.

The result of applying two times the production which breaks the mesh interior, two times the production which breaks the north edge, two times the production which breaks the south edge, one time the production which breaks the west edge, one time the production which breaks the east edge, and one time the production which breaks the shared edge to the hypergraph from Fig. 4b is shown in Fig. 13.

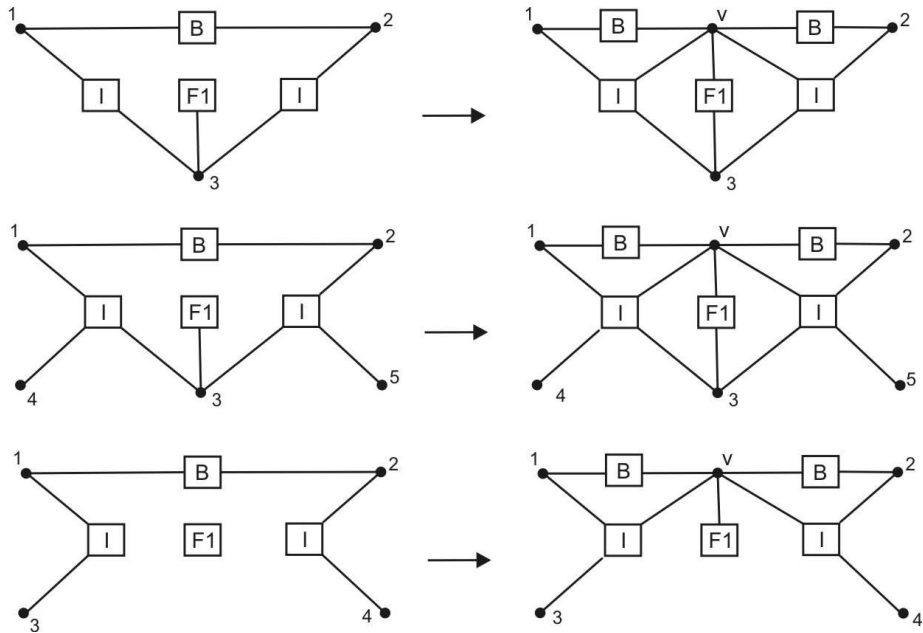


Fig. 10. Hypergraph productions for breaking boundary edges

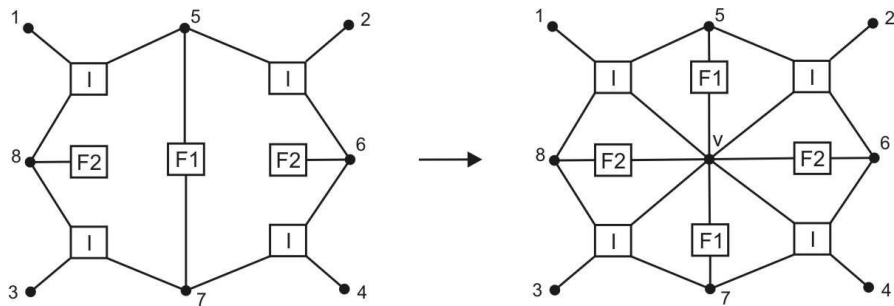


Fig. 11. A hypergraph production for breaking the shared edge

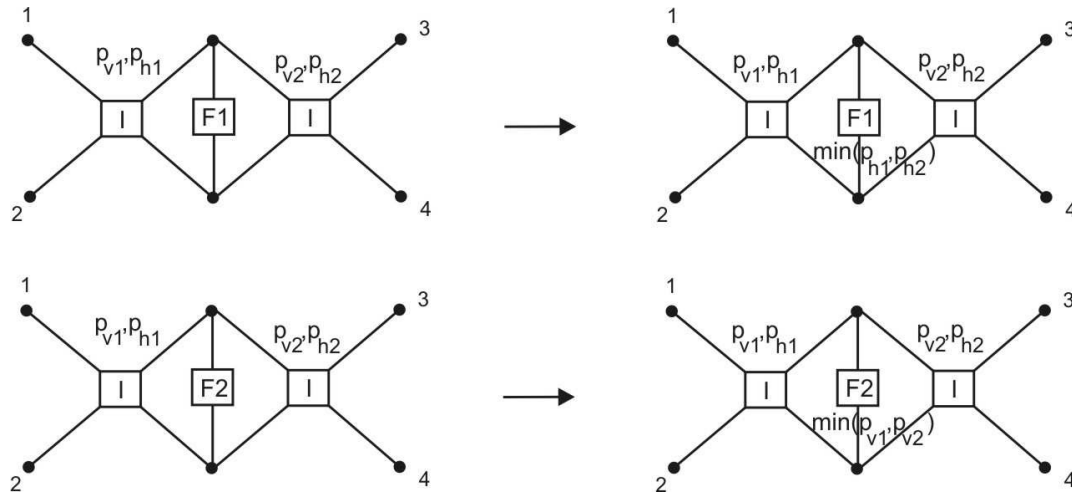


Fig. 12. A hypergraph production for assigning appropriate polynomial orders of approximation for edges

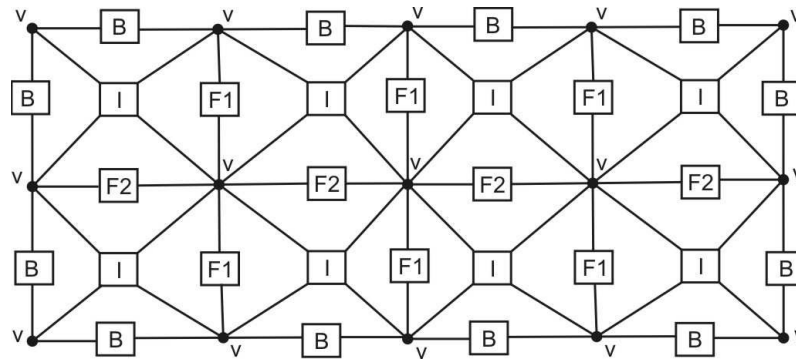


Fig. 13. A hypergraph resulting after using productions breaking mesh elements to all hyperedges of the hypergraph from Fig. 4b.

4. Conclusions

The paper presents the hypergraph grammar for modelling the hp-adaptive finite element method algorithm with rectangular elements. The finite element mesh is represented by a hypergraph. All mesh transformations are modelled by means of hypergraph grammar productions. The hypergraph grammar allows one to generate the initial mesh, to assign values of polynomial order of approximation to the element vertices, edges and interiors, to generate the matrix for each element, to solve the problem, to make the virtual hp-adaptation as well as to perform the hp-adaptation.

The future work will concentrate on automatization of the derivation process of the hypergraphs representing the finite element mesh by defining the hypergraph grammar system. Moreover, computational complexity of the hypergraph grammar-driven hp-adaptive finite element method algorithm will be studied and compared with the complexity of the one which is composition graph grammar-driven.

Acknowledgements

The work presented in this paper was partially supported by Polish National Science Center grant no. NN 519447739.

References

- [1] Hughes TJR, 2000. Linear static and dynamic finite element analysis. Dover Publications.
- [2] Demkowicz L, 2006. Computing with Hp-Adaptive Finite Elements, Vol. 1: One and Two Dimensional Elliptic and Maxwell Problems. Chapman & Hall / CRC Press.
- [3] Demkowicz L, Kurtz J, Pardo D, Paszyński M, Rachowicz W, Zdunek A, 2007. Computing with Hp-Adaptive Finite Elements, Vol. 2: Frontiers: Three Dimensional Elliptic and Maxwell Problems with Applications. Chapman & Hall / CRC Press.
- [4] Flasiński M, Schaefer R, 1996. Quasi context sensitive graph grammars as a formal model of FE mesh generation, *Computer-Assisted Mechanics and Engineering Science*; 3:191-203.
- [5] Habel A, Kreowski HJ, 1987. May We Introduce to You: Hyperedge Replacement. *Lecture Notes in Computer Science*; 291:5–26.
- [6] Habel A, Kreowski HJ, 1987. Some Structural Aspects of Hypergraph Languages Generated by Hyperedge Replacement. *Lecture Notes in Computer Science*; 247:207–219.
- [7] Paszyński M, Paszyńska A, 2008. Graph transformations for modeling parallel hp-adaptive finite element method, *Lecture Notes in Computer Science*; 4967:1313-1322.
- [8] Paszyńska A, Grabska E, Paszyński M, 2012. A Graph Grammar Model of the hp Adaptive Three dimensional Finite Element Method, Part I. *Fundamenta Informaticae*; 114(2):149-182.
- [9] Paszyńska A, Grabska E, Paszyński M, 2012. A Graph Grammar Model of the hp Adaptive Three dimensional Finite Element Method, Part II. *Fundamenta Informaticae*; 114(2):183-201.
- [10] Spicher A, Michel O, Giavitto J, 2010. Declarative Mesh Subdivision Using Topological Rewriting in MGS, *International Conference on Graph Transformation*, Enschede, The Netherlands, September 2010, *Lecture Notes in Computer Science*; 6372:298-313.